Computer Generated Forces' Realism Enhancement

N. Abdellaoui, M. Bourassa, F. Lichacz
DRDC Ottawa
3701 Carling Avenue
Ottawa, Ontario K1A-0Z4
CANADA

G. Parkinson
SIMFRONT
2835 Ashton Station Road
Ashton (Ottawa), Ontario K0A-1B0
CANADA

B. Cain
DRDC Toronto
1133 Sheppard Avenue West
Toronto, Ontario M3M-3B9
CANADA

Nacer.abdellaoui@forces.gc.ca / Mike.bourassa@forces.gc.ca / gparkinson@xplornet.com / Fred.lichacz@forces.gc.ca / Brad.cain@forces.gc.ca

ABSTRACT

Simulation is a promising technology to prepare for a world of uncertainty, to acquire skill or to study alternatives in which agents do not always play by the rules. Computer Generated Forces (CGFs) systems are the cornerstone of constructive simulations and are adequately designed for the symmetric mindset and well adapted to the Cold War era, if one assumes that all forces will act according to Standard Operating Procedures (SOPs). Although experts are guided by SOPs, SOPs are seldom followed to the letter because of the intractable nature of preconceiving every eventuality. This is compounded further in the fourth generation, non-kinetic warfare, exacerbating the inadequacy of current CGF systems. One of the key drawbacks of existing CGF systems is the lack of adequate representation of human influences such as perception, reasoning, decision making, or what is recognized here as a lack of Artificial Intelligence (AI).

A comparative analysis about AI capabilities in CGFs, concluded that AI capabilities are very limited and recommended the realisation of a complementary AI component that should operate with existing CGFs to overcome these deficiencies..

To fulfil this realisation, we undertook an architectural study based on an engineering approach. Because it is important to be uniformly compatible with as many CGF systems as possible, we want to be aware of tools or systems, relevant to the Canadian Forces, which would justify fundamentally different methodological approaches in implementation. The primary fulfilment of this work is the well-architected design with methodological building blocks that takes into consideration the appropriate architecture that suits the current and future CGF problem space.

1.0 INTRODUCTION

Simulation is exploited for a variety of purposes within both the defence and homeland security communities. The Computer Generated Forces (CGFs) systems are the cornerstone of constructive simulations and an efficient way of providing extra players in a synthetic environment containing human participants. They are a viable alternative in experimentation, concept analysis and development, tactics development, and training. They provide the capability to model hundreds and thousands of synthetic entities of various types in a

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.		
1. REPORT DATE **OCT 2010**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Computer Generated Forces Realism Enhancement**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **DRDC Ottawa 3701 Carling Avenue Ottawa, Ontario K1A-0Z4 CANADA**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S)
		11. SPONSOR/MONITOR’S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release, distribution unlimited**		
13. SUPPLEMENTARY NOTES **See also ADA564696. Human Modelling for Military Application (Applications militaires de la modelisation humaine). RTO-MP-HFM-202**		
14. ABSTRACT **Simulation is a promising technology to prepare for a world of uncertainty, to acquire skill or to study alternatives in which agents do not always play by the rules. Computer Generated Forces (CGFs) systems are the cornerstone of constructive simulations and are adequately designed for the symmetric mindset and well adapted to the Cold War era, if one assumes that all forces will act according to Standard Operating Procedures (SOPs). Although experts are guided by SOPs, SOPs are seldom followed to the letter because of the intractable nature of preconceiving every eventuality. This is compounded further in the fourth generation, non-kinetic warfare, exacerbating the inadequacy of current CGF systems. One of the key drawbacks of existing CGF systems is the lack of adequate representation of human influences such as perception, reasoning, decision making, or what is recognized here as a lack of Artificial Intelligence (AI). A comparative analysis about AI capabilities in CGFs, concluded that AI capabilities are very limited and recommended the realisation of a complementary AI component that should operate with existing CGFs to overcome these deficiencies.. To fulfil this realisation, we undertook an architectural study based on an engineering approach. Because it is important to be uniformly compatible with as many CGF systems as possible, we want to be aware of tools or systems, relevant to the Canadian Forces, which would justify fundamentally different methodological approaches in implementation. The primary fulfilment of this work is the well-architected design with methodological building blocks that takes into consideration the appropriate architecture that suits the current and future CGF problem space.**		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**14**	

constructive simulation. Existing CGFs are adequately designed for the symmetric mindset and well adapted to the Cold War era, yet they are inadequate for fourth generation warfare. One of the key drawbacks of existing CGF systems is the lack of adequate representation of human influences such as perception, reasoning, decision making, or what is recognized as a lack of Artificial Intelligence (AI) [1].

The overarching project of this present paper is exploring the integration of AI with CGF software through the development of an AI Module. Based on a comparative analysis about AI capabilities in CGFs [8][13], it proposes to address the AI gap in an interoperable fashion (i.e. the built capability will be reusable across different CGFs). The objective is not to replace the AI that exists in the current CGF applications, but to complement the CGF with superior decision making capabilities that will lead to improved autonomy and realistic behaviour of synthetic entities. This AI Module will provide new AI capabilities to CGF systems in order to improve the decisions making of synthetic entities. It will be built on a new "needs-based" agent architecture that closely models a human psychological framework and will be built using established standards for multi-agent systems with dynamic modules. The AI Module will also build upon the High Level Architecture (HLA) standards to share control of synthetic entities represented in the CGF and to define a common interface that any HLA-compliant CGF can be adapted to.

2.0 PAPER OVERVIEW

The key criterion for the acceptance of any product, process, or service resides in its least disturbance to the existing products that it will operate with, and its quasi-seamless integration with these products. In other words, the central problem to be solved is that of interoperation with existing systems. A proper architecture of the technical solution will create an opportunity for reuse, hence contributing to the addressing of the interoperability need.

The architecture of the AI Module is the cornerstone for the component's usability for CGFs and other simulations. Therefore, addressing architectural concerns in the application of AI to synthetic entities that will make it feasible to integrate the AI Module with other simulations, CGFs, federates, etc. needs to be considered first and foremost. These concerns include the architectural structure, the availability of interfaces, and any new tools and processes needed for the technical solution.

This paper presents an overview of the architectural component of the technical solution of the AI Module. This technical solution will equip synthetic entities with more realistic and autonomous behaviour. It will also provide a suitable interface such that the same AI can be used with different CGF applications. There are three components to the solution, the AI component, the HLA communications architecture, and the CGF interface. A conceptual drawing of the proposed solution is presented in Figure 1.

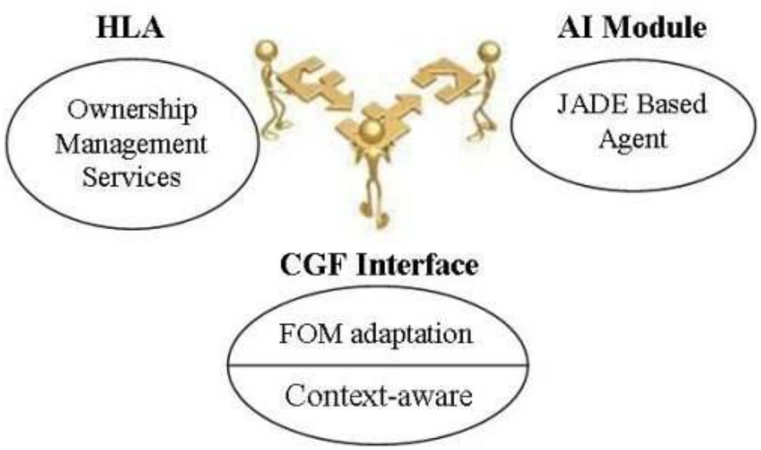

Figure 1: Concept of the proposed solution.

3.0 ARCHITECTURE

An architecture is a high level model of a complex system consisting of a detailed description of the system's components and the relationships among those components. It must have a specific stated purpose that guides its scope, development, and identifies its use. It provides a framework for evolving, maintaining, and integrating existing as well as emerging systems and processes. Architectures are not developed in a vacuum, and will more likely need to be integrated with other existing architectures.

The process of developing architectures starts with a thorough assessment of existing systems in order to bridge gaps, improve processes (automation, information flows' enhancement, etc), and anticipate the future. Simply put, architecture development is the first and most important step in implementing the problem's solution.

A stated purpose must be bound by a statement of scope and clarified through a scenario. The scenario selected for this research project consists of two vignettes:

- Maritime Patrol Aircraft patrolling a busy and dense region where the optimization and learning features will be exercised; and
- MANPADS targeting a helicopter where the application of initiative prediction will be evaluated.

Choosing a scenario helps scope the scale of the architecture development effort. This scale will determine the level of effort and investment required to complete the architecture. Although the architecture is not tailored toward the proposed scenarios, it will always consider it as an important use-case.

Architecturally, three components compose the solution's building blocks: the AI component, the HLA communications architecture, and the CGF interface. The AI component is a multi-agent system that incorporates a "needs-based" agent architecture operating in a Java Agent Development Framework (JADE) container with Open System Gateway Initiative (OSGi) extensions that provide a set of core services and resources [5]. The implementation of the multi-agent system is based on the psychological model Maslow's Hierarchy of Needs (MHN). HLA communications extend the existing simulation by utilizing the ownership management services provided by the Run Time Infrastructure (RTI) and defining any additional shared

information through FOM extensions. The CGF interface is a two fold capability; it extends the CGF's existing HLA capabilities to address the publication and subscription of the essential information defined in the FOM extensions; and it incorporates a context-aware agent-based component that will allow the passing of necessary environmental knowledge from the synthetic environment to the AI component [7]. The following three sections will address the main components of this architecture, in the following order: the HLA communications architecture, the AI component, and the CGF interface (Figure 2). The CGF interface is a two fold section; the HLA/FOM extensions and the context-aware agent.

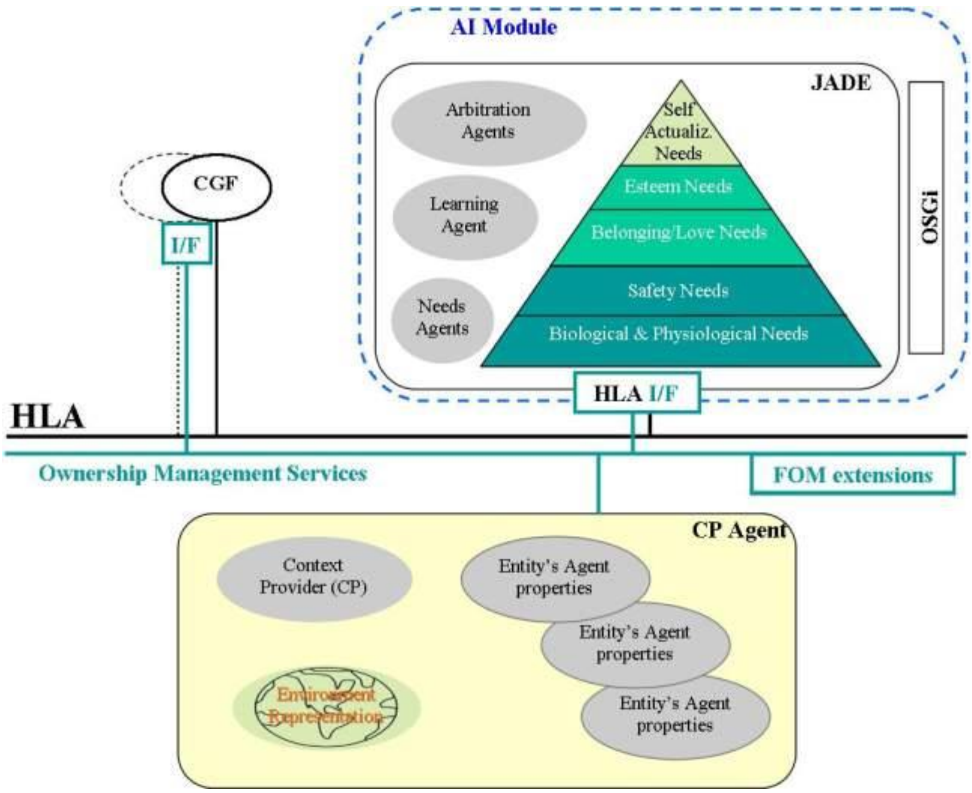

Figure 2: Concept Architecture for the AI in CGF project.

4.0 HIGH LEVEL ARCHITECTURE (HLA) OWNERSHIP MANAGEMENT SERVICES (OWMS)

The main idea behind the High Level Architecture (HLA) is to allow systems to interoperate on the implementation level. Under HLA, several distributed and heterogeneous components are composed using a common set of services delivered by a common software infrastructure for data exchange into one set of distributed simulations. The communication infrastructure of the HLA is built on a set of services that are collectively referred to as the Runtime Infrastructure (RTI).

In HLA nomenclature individual simulations/simulators are called federates and a collection of federates interacting in a joined simulation is referred to as a federation. Systems, employing HLA as the intercommunication mechanism, provide generic reusable solutions to support system management of

distributed simulations. The Object Model Template (OMT) describes the exchange of data between federates, and the RTI services are supporting the collaboration between these federates and their synchronization. Theoretically, the FEDEP assures the alignment of federate behaviour with respect to other federates. HLA has many years of existence, and is a mature enough method for coupling many federates within a federation and it can also offer interesting solutions if the HLA rules are fully exploited.

HLA offers a lot on the level of technical interoperability and technical connectivity which is a necessary requirement for the interoperation of systems. In particular the RTI services are a mature implementation of M&S requirements concerning parallel and distributed simulation using heterogeneous implementations. That said, HLA cannot solely address the AI gaps mentioned in the beginning of this paper since it is independent of a particular application and lacks any capability to provide context information to a simulation; nonetheless, its open architecture will contribute in externalising the CGF's gaps being addressed here.

The dynamic services of HLA; such as Time Management, Data Distribution Management, etc.; have also proven to be of great use to the M&S community. The dynamic aspect of special interest to this paper is the Ownership Management Services.

The original intent of the OWMS services was to allow for different "representational techniques to be selectively applied at different phases of a simulation application" (Dahmann 1999). OWMS are used in existing simulations to transfer simulation instance objects between low fidelity and higher fidelity federates as the simulation transitions through its various phases (Murphy et al. 2004). OWMS were designed to provide the flexibility of using a low fidelity model in a default way with the possibility of transiting to a high fidelity model when needed. Additional potential uses for OWMS include load balancing and fault recovery (Dahmann 1999). OWMS comprises functions supporting dynamic transfer of ownership of object attributes during the execution of the simulation, offering thus an increased flexibility to the HLA federation. Thus, the object's attributes (originally owned by the federate that instantiate the object) can be transferred to and managed by another federate following a well defined procedure. This procedure consists of either a "push" from the default owner or a "pull" from an unowning federate seeking to own that attribute. Ownership is tracked by the RTI and is not based on the instance, but rather on the individual attributes of the object instance [12].

Individual attributes that represent synthetic entity actuators can be shared between the CGF and the AI Module without the risk of collisions or conflicts since the RTI checks the ownership of object attribute instances thereby brokering who 'owns' the right to publish data on that attribute. But, since attribute-based ownership is not part of the DIS legacy on which RPR-FOM is based, ownership transfer may be implemented as a full entity transfer under strict time management to ensure updates are not missed. Consequently, the preferred implementation is for the AI Module to use the PULL transfer method (see Figure 3) to acquire ownership and retain ownership as long as needed. Once the AI Module is not needed to direct behaviour, the original parameters of the entity behaviour are restored and control is relinquished by an UNCONDITIONAL PUSH transfer (see Figure 4). It is assumed that the original owning federate will be polling the RTI looking to reacquire ownership since the HLA does not allow for a directed transfer of ownership to a specfic federate.

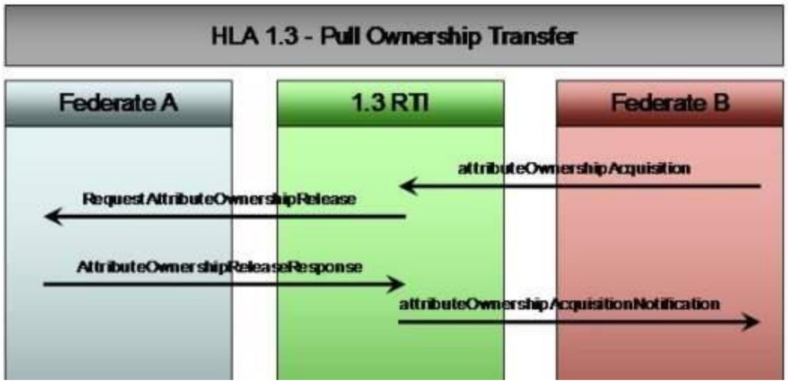

Figure 3: PULL Ownership Transfer.

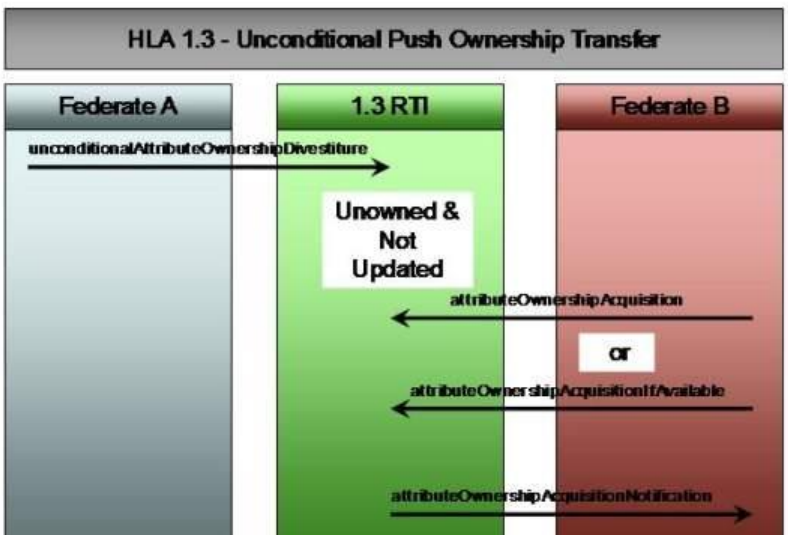

Figure 4: UNCONDITIONAL PUSH Ownership Transfer.

5.0 ARTIFICIAL INTELLIGENCE MODULE

The improvement of the synthetic entity behaviour derives from reinterpreting the entity's environmentally sensed data and proposing better actuator configurations. This implies that the entity sensor data and actuators are made available to the AI Module. There may also be other internal state attributes the AI needs to perform its evaluation and that information also needs to be made available to the AI Module. If an HLA mechanism is to be used, it is necessary to have these attributes defined in the FOM.

5.1 Agent Definition

An agent is a self-contained piece of software that perceives its environment through sensors; alters its state through actuators or effectors and communicates with its environment. The set of inputs is identified as a

Percept, and the output involving an effector is identified as Action. The agent can have goals which it tries to achieve or derive unobservable characteristics by analyzing percepts.

5.1.1 Agent's Categories

Agent categories vary from pure reflex agents to a more general agent framework [11]:

- Percept-based Agent is an agent that gets information from its sensors, changes its current state and triggers actions through the effectors. It is also identified as reactive agent. Reactive agents have no notion of history. The current state is as the sensors see it right now. The action is based on the current percepts only. Such agents are quite efficient, but not fit for multiple goals. They have no internal representation for reasoning, strategic planning or learning.
- State-Based Agent differs from the previous one in the fact that it maintains some sort of state based on the received percept. Hence, the actions it triggers are based on both the sensors and previous knowledge (memory).
- Goal-Based Agent has a goal which forms a basis of its actions. The actions it triggers are based, in addition to the other inputs, on goals and intentions. Goal formulation based on the current situation is a way of solving many problems and search is a universal problem solving mechanism in AI. The sequence of steps required to solve a problem is not known a priori and must be determined by a systematic exploration of the alternatives.
- Utility-Based Agent provides a more general agent framework. In case that the agent has multiple goals, this framework can accommodate different preferences for the different goals. Such systems are characterized by a utility function that maps a state or a sequence of states to a real valued utility. The agent acts towards maximizing expected utility.
- Learning Agent allows an agent to operate in initially unknown environments. The learning element modifies the performance element.

5.2 AI Module Framework

The AI Module consists of the JADE multi-agent framework operating within an OSGi framework. JADE provides an off-the-shelf framework for the implementation of a multi-agent system [6]. It includes capabilities for agent construction, the specification and management of behaviours, and for inter-agent communications including the specification of ontologies and communications protocols which are essential to achieve the self-organizational aspects of this project. It also provides infrastructure services for monitoring operation of the framework as well as libraries and support tools for development of agents [3]. The OSGi is a set of specifications that define a dynamic module system for Java-based systems. An OSGi framework allows for the creation and management of common components called 'bundles' with clear and strict boundaries [14]. OSGi provides the ability to run simultaneously multiple instances of the same version of an agent as well as multiple versions of the same agent [4].

The key capabilities of the AI Module are contained within the structure of the agents inside the JADE framework. In the "needs-based" architecture (based on the psychological framework known as MHN), agents are organized based on a hierarchical framework and model elements of each layer in the framework. Each layer in the MHN framework will consist of one or more agents. Agents receive information from explicit sensors or from agents at lower levels. Agents at each level propose goals which are then arbitrated by level and finally arbitrated to a single set of actuator commands for implementation [15]. The trend in the arbitration will favour the lower levels, but the possibility will exist that higher goals will trump lower level

needs. This framework can operate both at the level of an individual entity or a formation of entities. In the case where a formation and an entity are both modelled in this fashion, the formation goals and objectives would be captured in the higher levels of the entity model. The internal agent structure will follow the template described in "Agent Architecture" section below.

5.3 Agent Architecture

The Agent architecture is a new and novel construct for this project. In this architecture a multi-agent system is created paralleling the MHN model. This construct specifies three types of agents (see Figure 6). Pyramid agents are those that address part (or all) of one level of the MHN pyramid. Each agent is designed following the structure presented in Figure 5. Arbitration agents are used to resolve conflicts in agent goals and present a single control solution for implementation. Learning agents are employed to recognize when expected states do not meet actual states and dynamically adapt the agent accordingly. Learning occurs as a cyclic process of evaluating the gained utility of actions/needs against the world model. Learning agents are used to refine the parameters in either the Pyramid Agents or the Arbitration Agents. They can be used to add/modify behaviours or change the structure of the behaviours accomplished by an agent. The implementation of learning is a key requirement to achieve truly autonomous operation of synthetic entities.

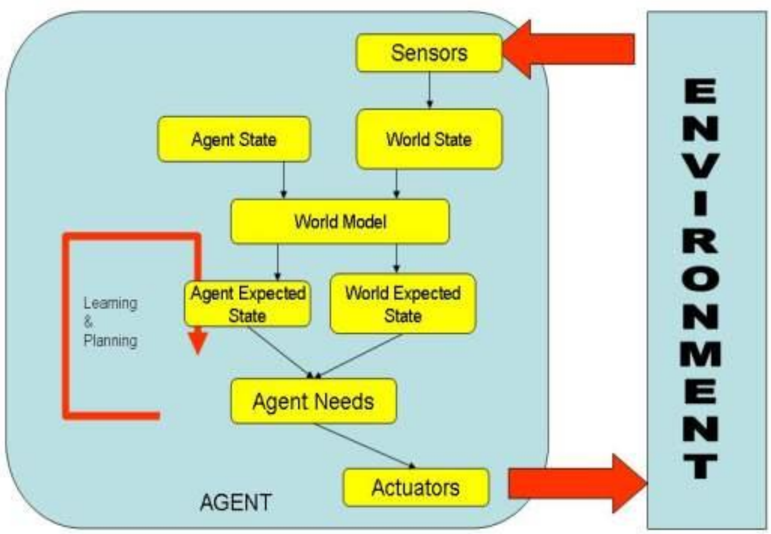

Figure 5: Pyramid Agent Structure.

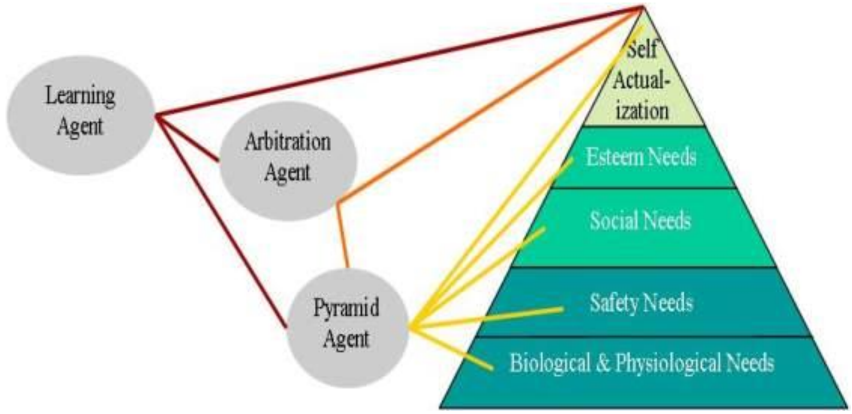

Figure 6: Agent Architecture using MHN.

6.0 COMPUTER GENERATED FORCES INTERFACE

As mentioned previously, for the AI Module to augment decision making and to improve synthetic entity behaviour, requires synthetic environment entity information which can be provided via the HLA federation. This implies that certain attributes may be added or refined in the FOM. It will also require environment information such as terrain, features, roads, obstacles, etc. To be able to acquire the environment data, this work is leveraging from an approach used in developing context-aware applications.

6.1 FOM Extension

FOM extensions define the scope of state, sensor and actuator information/controls to operate in an HLA federation. Using the HLA to communicate this information activates the concept of ownership of information. In other words, the HLA will not allow information to be published except by the federate that possesses ownership of that information and the RTI will actively enforce that rule. It is expected that in most cases, the AI Module will be added to an existing federation instead of creating a new federation. If the additional information needed for the AI Module in the FOM is solely additions to the existing FOM, then the remainder of the federation will operate in blissful ignorance of the FOM extensions. The result will be that federates not subscribing to the AI Module will not be impacted by the AI Module's presence. This is the preferred approach. Within an existing FOM, the demarcation structures may already exist or require additional attributes/parameters to be exposed in order to have the synthetic entities operate with the AI Module, or will be implemented in elaborations of existing attributes/parameters such as those of the Real-time Platform Reference Federation Object Model (RPR-FOM).

6.2 Context Awareness

The context-aware concept helps discern relevant from non-relevant context and how to help agents make appropriate decisions based on it.

6.2.1 Context for AI Module

Context is defined as any information that is considered relevant to the agent action (e.g. agent's position, identities [of SE entities] around the requesting agent's surrounding environment, routes, moving methods,

time of day, or weather information). Nevertheless, these are more special context information than a general definition. We are proposing in this sense an architecture that will allow the AI agent to receive the context information from a central Context Provider (CP) which holds the current ground truth of the SE. The role of the CP is to provide the agent with the set of needed context attributes. Therefore, the AI agent will deal directly with the CP for context information (see Figure 7).

Two categories on context are considered here:

- The SE entity's surrounding world which we can identify as environmental context, which is common to all synthetic entities, and
- The other SE entities surrounding and interacting with the concerned SE entity.

6.2.1.1 Context Provider (CP)

The main functionality of the Context Provider (CP) is to compute a context attributes set as per the agent request. CP is responsible of maintaining an up to date environmental and entities list database. Three mechanisms are available to the CP to stay up to date:

- Initial loading of the environment (terrain, routes, features, etc)
- Periodical access to the information that changes constantly (position, orientation, etc), and
- Per status change loading (health, weather, etc).

6.2.1.2 Specification of Context Information

This concept is borrowed to the "Creation of User-friendly Mobile services Personalised for Tourism" (CRUMPET) project, where context information is processed, evaluated and passed on using GML (Geographic Markup Language) [9]. CRUMPET also uses the location-related language POIX (point of Interest eXchange Language) of the W3C (World Wide Web Consortium) to exchange location-related information.

The Geographic Markup Language (GML)

GML is a XML representation of Simple Features. In order to draw a map it is necessary to transform the GML into a graphic format, either by direct rendering, or by transformation into (XML encoded) graphics elements. This can be done anywhere in the processing chain between the data store and the visualization device. GML can be related to other new XML-based standards like the "Point Of Interest eXchange Language" (POIX) from the W3C Consortium. This is a more simplified model for position and direction information. POIX data can be generated from GML [2].

Point Of Interest eXchange Language (POIX)

Point Of Interest eXchange Language (POIX) is a location-related information descriptive language prepared with the aim of exchanging location-related information over the Internet, and is designed with XML 1.0* (Extensible Markup Language [W3C Recommendation]). Not only does POIX denote a simple location, but it also provides an environment capable of representing comprehensive information associated with the targeted location [10].

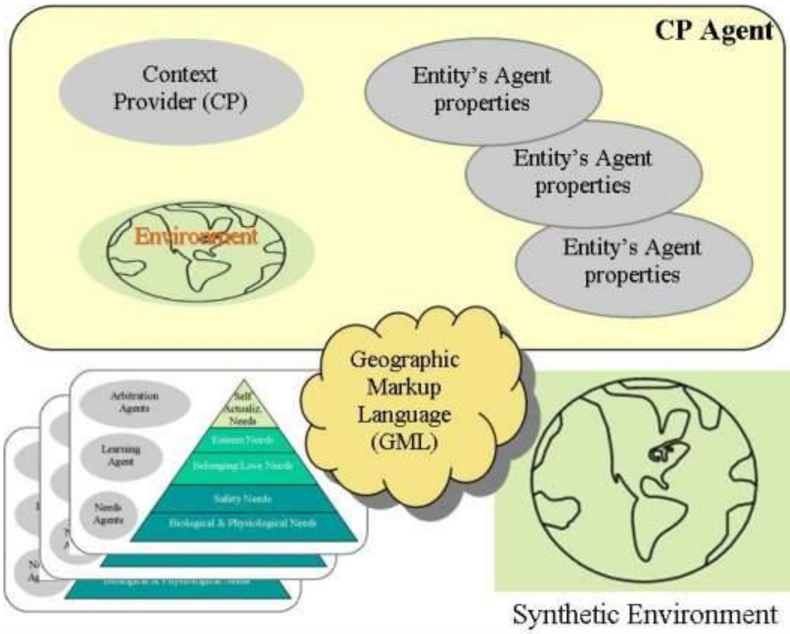

Figure 7: Architecture of Context-Provider Agent.

7.0 CONCLUSION

Existing CGF systems present some fundamental and very critical limitations whose solution may benefit from the concepts and techniques of AI. The reasoning models in these systems are quite limited and mostly rule-based, which were adequate for what these systems were initially created for. However, the fourth generation warfare disrupted many of the cold war era concepts. Conceptual models of existing CGFs didn't escape this disruption and need to adjust to the new warfare era. This project will contribute to the adjustment of the CGFs to the new needs from a perspective of an adequate representation of human influences.

The strategy described in this paper is a composability-based approach and is applicable for HLA-compliant CGFs and might be easily adaptable for other simulations that are HLA compliant. The adopted open architecture can accommodate newer AI requests from both individual entities and formations.

Reusability is a key building-block of composability and is also a concept that this project is adopting. Yet, typically reusable models are initiated as components in a larger system and must interact adequately within the larger system and might be tailored in a non-interoperable way; which might require extra effort to reuse them elsewhere. From a pure software development perspective, software disciplines are successfully applying component-based approach to build software systems, the challenge in our specific project might come from:

- The CGF systems, even though, HLA compliant, might lack the necessary dynamics (composability, openness, modularity, etc) to seamlessly embed these models; and
- A model is only reusable to the extent that its original system's assumptions are consistent with the constraints of the new CGF system. Without appropriate information to guide the adaptation, a model may not be reused advantageously within a new system.

Changes to the FOM might imply internal changes to the CGF. There may be new publication/subscription structures that need to be implemented as well as ties between the HLA data and any internal data structures within the CGF. This also may require additional data publishing steps that need to be captured in the main simulation loop or support for update value requests, which otherwise would not be supported. The HLA specification provides all the necessary services, but they may not all be implemented within a specific CGF.

A potentially challenging situation that this project might face will come from the apparent environment, and serious thoughts will be given to the design and implementation of the context-aware agent framework.

We explored an approach to developing such an intelligent agent based on the HLA architecture framework, a hybrid methodology that combines elements of AI and context-aware agent. We proposed a concrete methodology by which the AI Module could engage the CGF based on the HLA ownership management services.

This is an architecture phase, and as such, many challenges in this interactions' representation still need to be verified or adjusted along the project's development progress. The leveraging being exposed here from other domains might come across matters that will require adjustment to the design of the component. Nevertheless, the architecture as laid down here is taking all these aspects into consideration.

REFERENCES

[1] Abdellaoui, N., Taylor, A., Parkinson, G. "Comparative Analysis of Computer Generated Forces' Artificial Intelligence" Symposium Proceedings RTO-MP-MSG-069, Paper 2. Brussels, Belgium, October 2009

[2] Cuthbert, A. et al, "Geography Markup Language (GML) v1.0" - OGC Document Number: 00-029, May 2000.

[3] Jade Programmer's Guide – JADE online documentation, 2010 available at http://jade.tilab.com/doc/index.html

[4] JADE OSGi Guide, JADE online documentation, 2010 available at http://jade.tilab.com/doc/index.html

[5] JADE Gateway tutorial, JADE online documentation, 2010 available at http://jade.tilab.com/doc/index.html

[6] JADE Administrator's Guide, online documentation, 2010 available at http://jade.tilab.com/doc/index.html

[7] Oana, B., Philippe, B., Olivier, B. 2005. "Representing Context in an Agent Architecture for Context-Based Decision Making. In Proceedings of CRR'05 Workshop on Context Representation and Reasoning, July 5, Paris, France.

[8] Parkinson, G., Abdellaoui, N., Taylor, A. "AI in CGF Comparative Analysis" – Summary Report – DRDC Internal Report - Ottawa, 2009.

[9] Poslad, S., Laamanen, H., Malaka, R., Nick, A., Buckle, P., Zipf, A.: CRUMPET: Creation of User-Friendly Mobile Services Personalised for Tourism. 3G Mobile Communication Technologies. London, UK (2001) 26-29

[10] POIX: Point Of Interest eXchange Language Specification-W3C Note, http://www.w3.org/TR/poix/ , accessed Aug 2010.

[11] Russell S., Norvig, P. "Artificial Intelligence: A Modern Approach" 3rd Edition –. Prentice-Hall Inc. New Jersey, 2010.

[12] SISO-REF-032-2009, Transfer of Control Study Group Final Report, available from http://www.sisostds.org/index.php?tg=articles&idx=More&article=40&topics=18 , accessed July 2010.

[13] Taylor, A., Abdellaoui, A., Parkinson, G., "Artificial Intelligence in Computer Generated Forces", HSC Proceedings of the Huntsville Simulation Conference 2009, Huntsville, Alabama, October 2009.

[14] Watson, T., Kaegi S., Webinar: Introduction to Eclipse Equinox and OSGi available at http://live.eclipse.org/node/387, accessed 26 July 2010.

[15] Wooldridge, M., "An Introduction to Multi-agent Systems", John Wiley Ed., 2002.